

**REMARKS**

In the non-final Office Action, the Examiner objected to claims 1, 7, 8, 10, 13, 14, 18-20, 24, and 25 due to minor informalities; rejected claims 1, 7, 12, and 13 under 35 U.S.C. § 112, second paragraph, as indefinite; rejected claims 1, 2, 4-7, 10-13, and 20-25 under 35 U.S.C. § 102(b) as anticipated by Mattis et al. (U.S. Patent No. 6,209,003); rejected claim 3 under 35 U.S.C. § 103(a) as unpatentable over Mattis et al. in view of Manley et al. (U.S. Patent Application Publication No. 2003/0182330); and rejected claims 8, 9, and 14-19 under 35 U.S.C. § 103(a) as unpatentable over Mattis et al. in view of Hisgen et al. ("New-Value Logging in the Echo Replicated File System," June 23, 1993).

By this Amendment, Applicants amend claims 1, 7-14, 16, 18-22, 24, and 25 to improve form. Applicants respectfully traverse the Examiner's objections and rejections under 35 U.S.C. §§ 112, 102, and 103. Claims 1-25 remain pending.

***CLAIM OBJECTIONS***

In paragraph 4 of the Office Action, the Examiner objected to claims 1, 7, 8, 10, 13, 14, 18-20, 24, and 25 for minor informalities. Applicants have amended the claims to address these informalities. Accordingly, Applicants respectfully request that the objections to the claims be reconsidered and withdrawn.

***REJECTION UNDER 35 U.S.C. § 112, SECOND PARAGRAPH***

In paragraph 6 of the Office Action, the Examiner rejected claims 1, 7, 12, and 13 under 35 U.S.C. § 112, second paragraph, as allegedly indefinite for failing to particularly point out and distinctly claim that which Applicants regard as the invention. The Examiner alleged that the phrase "do not exist" renders the claims indefinite. Applicants have amended the claims to

remove this phrase. Accordingly, Applicants respectfully request that the rejection of claims 1, 7, 12, and 13 under 35 U.S.C. § 112, second paragraph, be reconsidered and withdrawn.

*REJECTION UNDER 35 U.S.C. § 102 BASED ON MATTIS ET AL.*

In paragraph 8 of the Office Action, the Examiner rejected claims 1, 2, 4-7, 10-13, and 20-25 under 35 U.S.C. § 102(b) as allegedly anticipated by Mattis et al. Applicants respectfully traverse the rejection.

A proper rejection under 35 U.S.C. § 102 requires that a single reference teach every aspect of the claimed invention either expressly or impliedly. Any feature not directly taught must be inherently present. In other words, the identical invention must be shown in as complete detail as contained in the claim. See M.P.E.P. § 2131. Mattis et al. does not disclose or suggest the combination of features recited in claims 1, 2, 4-7, 10-13, and 20-25.

For example, amended independent claim 1 is directed to a method for deleting one or more of a plurality of files, where the files include one or more chunks stored by a plurality of servers. The method comprises identifying a file to be deleted; renaming the identified file; permanently deleting the renamed file a predetermined amount of time after renaming the identified file as part of a garbage collection process; receiving, from the servers, information concerning chunks stored by the servers; and identifying, to one of the servers, one of the chunks that corresponds to the permanently deleted file.

Mattis et al. does not disclose or suggest the combination of features recited in claim 1. For example, Mattis et al. does not disclose or suggest renaming a file that is identified to be deleted. The Examiner alleged that Mattis et al. discloses this feature and cited column 3, lines 16-19, of Mattis et al. for support (Office Action, page 3). Applicants respectfully disagree.

At column 3, lines 16-19, Mattis et al. discloses:

Examples include: support for random access and selective modification, file permissions, support for moving files, support for renaming files, and support for appending to files over time.

This section of Mattis et al. discloses that a prior art approach to structure cache object stores includes support for renaming files. While this section of Mattis et al. discloses that files may be renamed for purposes of structuring cache object stores, this section of Mattis et al. falls short of disclosing renaming a file that is identified to be deleted, as required by claim 1. Instead, Mattis et al. discloses that if a fragment is to be deleted, then it is deleted by marking it as deleted and overwriting the data in the fragment (col. 23, lines 15-17). Nowhere does Mattis et al. disclose or suggest renaming a file that is identified to be deleted, as required by claim 1.

Further, even if, for the sake of argument, this section of Mattis et al. could be equated to renaming a file that is identified to be deleted (a point that Applicants do not concede), the Examiner has identified a prior art approach as allegedly equivalent to this feature of claim 1 but relied upon the system of Mattis et al. for other features of the claim 1. The Mattis et al. system does not include this prior art approach and the Examiner has not provided the requisite motivation for modifying the Mattis et al. system to include this prior art approach. Accordingly, the Examiner has not established a prima facie basis to deny patentability with regard to claim 1.

Mattis et al. also does not disclose or suggest permanently deleting the renamed file a predetermined amount of time after renaming the identified file as part of a garbage collection process. The Examiner alleged that Mattis et al. discloses this feature and cited column 16, lines 48-52, column 21, lines 52-55, and column 22, lines 43-47, of Mattis et al. for support (Office Action, page 3). Applicants respectfully disagree.

At column 16, lines 48-52, Mattis et al. discloses:

For example, the Open Directory is useful in safeguarding against overwriting or deleting an object that is currently being read. The Open Directory also buffers changes to the Directory Table 110 before they are given permanent effect in the Directory Table 110.

In this section, Mattis et al. discloses that the Open Directory buffers changes to the Directory Table before they are given permanent effect. Nowhere in this section, or elsewhere, does Mattis et al. disclose or suggest permanently deleting a renamed file a predetermined amount of time after renaming a file that is identified to be deleted as part of a garbage collection process, as required by claim 1.

At column 21, lines 52-55, Mattis et al. discloses:

Preferably, the garbage collection method is implemented as an independent process that runs in parallel with other processes that relate to the cache. This enables the garbage collection method to periodically clean up cache storage areas without interrupting or affecting the operation of the cache.

In this section, Mattis et al. discloses that a garbage collection method periodically cleans up the cache storage areas. Nowhere in this section, or elsewhere, does Mattis et al. disclose or suggest permanently deleting a renamed file a predetermined amount of time after renaming a file that is identified to be deleted as part of a garbage collection process, as required by claim 1.

At column 22, lines 39-47, Mattis et al. discloses:

In step 806, one of the fragments within the selected arena is selected for garbage collection. In determining which fragment or fragments to select, the cache 80 takes into account several selection factors, as indicated by block 807. In the preferred embodiment, the factors include: the time of the last access to the fragment; the number of hits that have occurred to an object that has data in the fragment; the time required to download data from the fragment to a client; and the size of the object of which the fragment is a part.

In this section, Mattis et al. discloses factors that are taken into consideration when determining which fragment to select for garbage collection. Nowhere in this section, or elsewhere, does

Mattis et al. disclose or suggest permanently deleting a renamed file a predetermined amount of time after renaming a file that is identified to be deleted as part of a garbage collection process, as required by claim 1.

Moreover, claim 1 recites additional features that are neither disclosed nor suggested in Mattis et al. For example, Mattis et al. does not disclose or suggest identifying, to one of the servers, one of the chunks that corresponds to the permanently deleted file, as further recited in claim 1. The Examiner alleged that Mattis et al. discloses this feature and cited column 33, lines 32-40, of Mattis et al. for support (Office Action, page 4). Applicants respectfully disagree.

At column 33, lines 32-40, Mattis et al. discloses:

If no match of the key is found in the search, then in step 848 the process returns an error message to the calling program or process, indicating that the requested object does not exist in the cache. Although the specific response to such a message is determined by the calling program or process, in the World Wide Web context, generally the proxy 30 contacts the server 40 that stores the object using an HTTP request, and obtains a copy of the requested object.

In this section, Mattis et al. discloses that a search is performed for a key to determine whether a requested object exists in the cache and if no match of the key is found, the proxy contacts the server to obtain a copy of the requested object. Nowhere in this section, or elsewhere, does Mattis et al. disclose or suggest identifying, to one of the servers, one of the chunks that corresponds to a file that was permanently deleted a predetermined amount of time after renaming the file, as required by claim 1.

For at least these reasons, Applicants submit that claim 1 is not anticipated by Mattis et al. Claims 2, 4-7, 10, and 11 depend from claim 1 and are, therefore, not anticipated by Mattis et al. for at least the reasons given with regard to claim 1. Claims 2, 4-7, 10, and 11 are also not

anticipated by Mattis et al. for reasons of their own.

For example, claim 4 recites that the predetermined amount of time is a user-configurable amount of time. As explained above with regard to claim 1, Mattis et al. is completely silent with regard to permanently deleting a renamed file a predetermined amount of time after renaming the file. Therefore, Mattis et al. cannot disclose or suggest that the predetermined amount of time is a user-configurable amount of time, as required by claim 4. In other words, Mattis et al. is completely silent with regard to permanently deleting a renamed file a user-configurable amount of time after renaming the file, as required by claim 4.

The Examiner alleged that Mattis et al. discloses this feature and cited column 22, lines 14-23, of Mattis et al. for support (Office Action, page 4). Applicants respectfully disagree.

At column 22, lines 14-23, Mattis et al. discloses:

When the amount of active storage in a particular pool becomes greater than the "high water mark" value, garbage collection is initiated and carried out repeatedly until the amount of active storage in the pool falls below the "low water mark" value. The "low water mark" value is selected to be greater than zero, and the "high water mark" value is chosen to be approximately 20% less than the total storage capacity of the pool. In this way, garbage collection is carried out at a time before the pool overflows or the capacity of the storage device 90a is exceeded.

In this section, Mattis et al. discloses that garbage collection is initiated when the amount of active storage in a pool becomes greater than a high water mark. Nowhere in this section, or elsewhere, does Mattis et al. disclose or suggest permanently deleting a renamed file a user-configurable amount of time after renaming the file, as required by claim 4.

For at least these additional reasons, Applicants submit that claim 4 is not anticipated by Mattis et al. Claim 5 depends from claim 4 and is, therefore, also not anticipated by Mattis et al. for at least the reasons given with regard to claim 4.

Amended independent claims 12 and 13 recite features similar to, but possibly different in scope from, features recited in claim 1. Claims 12 and 13 are, therefore, not anticipated by Mattis et al. for at least reasons similar to reasons given with regard to claim 1.

Amended independent claim 20 is directed to a method for deleting stale replicas of chunks, where the replicas are stored by a plurality of servers. The method comprises associating version information with replicas of chunks; identifying stale replicas based on the associated version information; deleting the stale replicas; receiving, from the servers, information concerning replicas stored by the servers; and identifying, to one of the servers, one of the replicas that corresponds to one of the deleted stale replicas.

Mattis et al. does not disclose or suggest the combination of features recited in claim 20. For example, Mattis et al. does not disclose or suggest identifying and deleting stale replicas and identifying, to one of the servers, one of the replicas stored by the server that corresponds to one of the deleted stale replicas.

The Examiner alleged that Mattis et al. discloses identifying and deleting stale replicas and identifying, to the servers, ones of the replicas that are stale replicas and identified column 26, lines 15-22, of Mattis et al. for support (Office Action, page 6). Applicants respectfully disagree.

At column 26, lines 15-22, Mattis et al. discloses:

In block 1216, the cache updates an expiration date value stored in association with the information object to reflect the current date or time. By updating the expiration date, the cache ensures that the garbage collection process will not delete the object, because after the update it is not considered old. In this way, an old object is refreshed in the cache without retrieving the object from its origin, writing it in the cache, and deleting a stale copy of the object.

In this section, Mattis et al. discloses that the cache uses an expiration date value to determine whether to delete an object from the cache. Nowhere in this section, or elsewhere, does Mattis et al. disclose or suggest, for example, identifying, to one of the servers, one of the replicas stored by the server that corresponds to one of the deleted stale replicas.

For at least these reasons, Applicants submit that claim 20 is not anticipated by Mattis et al. Claims 21-23 depend from claim 20 and are, therefore, not anticipated by Mattis et al. for at least the reasons given with regard to claim 20.

Amended independent claims 24 and 25 recite features similar to, but possibly different in scope from, features recited in claim 20. Claims 24 and 25 are, therefore, not anticipated by Mattis et al. for at least reasons similar to reasons given with regard to claim 20.

*REJECTION UNDER 35 U.S.C. § 103 BASED ON MATTIS ET AL. AND MANLEY ET AL.*

In paragraph 10 of the Office Action, the Examiner rejected claim 3 under 35 U.S.C. § 103(a) as allegedly unpatentable over Mattis et al. in view of Manley et al. Applicants respectfully traverse the rejection.

Initially, claim 3 depends on claim 1. The disclosure of Manley et al. does not cure the deficiencies in the disclosure of Mattis et al. identified above with regard to claim 3. Therefore, claim 3 is patentable over Mattis et al. and Manley et al., whether taken alone or in any reasonable combination, for at least the reasons given with regard to claim 3. Claim 3 is also patentable over Mattis et al. and Manley et al. for reasons of its own.

Claim 3 recites receiving an un-deletion instruction regarding the file and restoring an original name to the file without permanently deleting the renamed file. Neither Mattis et al. nor Manley et al. discloses or suggests the combination of features recited in claim 3. For example,



neither Mattis et al. nor Manley et al. discloses or suggests restoring an original name to a renamed file without permanently deleting the renamed file.

The Examiner admitted that Mattis et al. does not disclose this feature, but alleged that Manley et al. discloses the feature and cited paragraph 0067 of Manley et al. for support (Office Action, page 7). Applicants respectfully disagree.

At paragraph 0067, Manley et al. discloses:

FIG. 7 shows an exemplary inode file system structure 700 after a file data block has been modified. In this illustrative example, file data which is stored at disk block 520C is modified. The exemplary WAFL file system writes the modified contents to disk block 520C', which is a new location on disk. Because of this new location, the inode file data which is stored at disk block (515) is rewritten so that it points to block 520C'. This modification causes WAFL to allocate a new disk block (715) for the updated version of the data at 515. Similarly, the inode file indirect block 510 is rewritten to block 710 and direct block 512 is rewritten to block 712, to point to the newly revised inode 715. Thus, after a file data block has been modified the snapshot inode 605 contains a pointer to the original inode file system indirect block 510 which, in turn, contains a link to the inode 515. This inode 515 contains pointers to the original file data blocks 520A, 520B and 520C. However, the newly written inode 715 includes pointers to unmodified file data blocks 520A and 520B. The inode 715 also contains a pointer to the modified file data block 520C' representing the new arrangement of the active file system. A new file system root inode 705 is established representing the new structure 700. Note that metadata in any snapshot blocks (e.g. blocks 510, 515 and 520C) protects these blocks from being recycled or overwritten until they are released from all snapshots. Thus, while the active file system root 705 points to new blocks 710, 712, 715 and 520C', the old blocks 510, 515 and 520C are retained until the snapshot is fully released.

In this section, Manley et al. discloses that after a file data block has been modified, the snapshot inode contains a pointer to the original inode file system indirect block which contains a link to the inode. Nowhere in this section, or elsewhere, does Manley et al. disclose or suggest restoring an original name to a renamed file without permanently deleting the renamed file, as required by claim 3.

For at least these reasons, Applicants submit that claim 3 is patentable over Mattis et al. and Manley et al., whether taken alone or in any reasonable combination.

*REJECTION UNDER 35 U.S.C. § 103 BASED ON MATTIS ET AL. AND HISGEN ET AL.*

In paragraph 11 of the Office Action, the Examiner rejected claims 8, 9, and 14-19 under 35 U.S.C. § 103(a) as allegedly unpatentable over Mattis et al. in view of Hisgen et al.

Applicants respectfully traverse the rejection.

Claims 8 and 9 depend from claim 1. Without acquiescing in the Examiner's rejection, Applicants submit that the disclosure of Hisgen et al. does not cure the deficiencies in the disclosure of Mattis et al. identified above with regard to claim 1. Therefore, claims 8 and 9 are patentable over Mattis et al. and Hisgen et al., whether taken alone or in any reasonable combination, for at least the reasons given with regard to claim 1.

Amended independent claim 14 is directed to a method for deleting orphaned chunks of a plurality of chunks stored by a plurality of servers. The method comprises providing a mapping of file names to chunks; identifying chunks, as orphaned chunks, that are not reachable from any of the file names; deleting the orphaned chunks; receiving, from the servers, information concerning chunks stored by the servers; and identifying, to one of the servers, one of the chunks that corresponds to one of the deleted orphaned chunks.

Neither Mattis et al. nor Hisgen et al., whether taken alone or in any reasonable combination, discloses or suggests the combination of features recited in claim 14. For example, neither Mattis et al. nor Hisgen et al. discloses or suggests identifying, to one of the servers, one of the chunks that corresponds to one of the deleted orphaned chunks.

The Examiner alleged that Hisgen et al. discloses identifying, to the servers, ones of the chunks that are orphaned chunks and cited page 24, paragraph 3, lines 4-5, of Hisgen et al. for support (Office Action, page 10). Applicants respectfully disagree.

At page 24, paragraph 3, Hisgen et al. discloses:

Orphan list. An orphan file is a file that is no longer in the name space, in that no entry in any directory points to it, but it is still open on some client machine and therefore must still be kept in existence. (We use the term “orphan” because an orphan file has no parent directory.) The orphan list lists all orphan files. When the Echo distributed client-server caching algorithm [21] reveals that nobody has an orphan file open, EchoBox can remove the file from its orphan list and delete the file, returning its pages to the disk space allocator and removing it from the fid map.

In this section, Hisgen et al. discloses that when an orphan file is not open, the file can be removed from the orphan list and deleted. Nowhere in this section, or elsewhere, does Hisgen et al. disclose or suggest identifying, to one of the servers from which information concerning chunks stored by the servers was received, one of the chunks that corresponds to one of the deleted orphaned chunks, as required by claim 14.

For at least these reasons, Applicants submit that claim 14 is patentable over Mattis et al. and Hisgen et al., whether taken alone or in any reasonable combination. Claims 15-17 depend from claim 14 and are, therefore, patentable over Mattis et al. and Hisgen et al. for at least the reasons given with regard to claim 14.

Amended independent claims 18 and 19 recite features similar to, but possibly different in scope from, features recited in claim 14. Claims 18 and 19 are, therefore, patentable over Mattis et al. and Hisgen et al., whether taken alone or in any reasonable combination, for at least reasons similar to reasons given with regard to claim 14.

#### *CONCLUSION*

In view of the foregoing amendments and remarks, Applicants respectfully request the Examiner's reconsideration of the application and the timely allowance of pending claims 1-25.

If the Examiner does not believe that all pending claims are now in condition for

allowance, the Examiner is urged to contact the undersigned to expedite prosecution of this application.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 50-1070 and please credit any excess fees to such deposit account.

Respectfully submitted,

HARRITY SNYDER, L.L.P.

By: 

Paul A. Harrity  
Reg. No. 39,574

Date: March 10, 2006

11350 Random Hills Road  
Suite 600  
Fairfax, Virginia 22030  
(571) 432-0800